Complex modulation computer-generated hologram with occlusion effect by a fast hybrid point-source/wave-field approach

Antonin Gilles, Patrick Gioia, Rémi Cozot, Luce Morin

Complex modulation computer-generated hologram with occlusion effect by a fast hybrid point-source/wave-field approach

Antonin Gilles[1*] Patrick Gioia[1,2] Rémi Cozot[1,3] Luce Morin[1,4]

[1] IRT b<>com Cesson-Sévigné France
[2] Orange Labs Rennes France
[3] University of Rennes 1 Rennes France
[4] INSA Rennes Rennes France

Abstract

We propose a fast Computer-Generated Hologram (CGH) computation method with occlusion effect based on a hybrid point-source/wave-field approach. Whereas previously proposed methods tried to reduce the computational complexity of the point-source or the wave-field approaches independently, the proposed method uses the two approaches together and therefore takes advantages from both of them. Our algorithm consists of three steps. First, the 3D scene is sliced into several depth layers parallel to the hologram plane. Then, light scattered by the scene is propagated and shielded from one layer to another, starting from the farthest layer. For each layer, light propagation and light shielding are performed using either a point-source or a wave-field approach according to a threshold criterion on the number of points within the layer. Finally, we compute light propagation from the nearest layer to the hologram plane in order to obtain the final CGH. Experimental results reveal that this combination of approaches does not produce any visible artifact and outperforms both the point-source and wave-field approaches.

Keywords: Computer-Generated Hologram, Color holography, Three-dimensional imaging

*This work has been achieved within the Institute of Research and Technology b<>com, dedicated to digital technologies. It has been funded by the French government through the National Research Agency (ANR) Investment referenced ANR-A0-AIRT-07. Authors can be reached at {antonin.gilles, patrick.gioia, remi.cozot, luce.morin}@b-com.com.

1 Introduction

Holography is often considered as the most promising 3D visualization technology, since it can provide the most authentic three-dimensional illusion to the naked eye [1]. Over the past decades, several methods have been proposed to generate Computer-Generated Holograms (CGH) of synthetic or existing scenes by simulating the propagation of light scattered by the scene towards the hologram plane. One of the most challenging feature in CGH techniques is the ability to properly take into account occlusions between objects in a scene. Occlusion processing in CGH techniques, also called light shielding, is the counterpart of hidden surface removal in Computer Graphics (CG). However, since CGH provides motion parallax, the occlusion effect in CGH implies that the visibility of objects changes according to the movement of the viewer. Two approaches are commonly used for CGH computation: the point-source and the wave-field approaches.

The point-source approach samples 3D scenes by a collection of self-luminous points, and calculates light propagation from the scene towards the hologram plane as the sum of spherical waves scattered by each point. Light shielding is usually treated as a visibility test to check the existence of obstacles between an object point and each sampling point in the hologram plane [2, 3]. This approach is very flexible and does not impose any restriction on the scene geometry. However, its computational com-

plexity is very high since it requires one calculation per point of the scene per pixel of the hologram. In order to reduce the computational complexity, several methods have been proposed, including pre-computed look-up tables [4,5], difference and recurrence formulas [6,7], image holograms [8], wavefront recording planes [9, 10], and using GPU hardware [11, 12] or special purpose hardware [13]. Additionally, it is also possible to reduce the number of visibility tests needed for light shielding by grouping sampling points in the hologram plane and/or grouping scene points [14, 15].

In the wave-field approach [16–18], 3D scenes are sliced into depth layers parallel to the hologram and light scattered by the scene is propagated from one layer to another using diffraction formulas such as the Angular Spectrum [19]. Light shielding is performed at each diffraction step by multiplying the light field by the binary silhouette mask of the current layer. The computation of the Angular Spectrum involves the use of the Fast Fourier Transform (FFT) algorithm twice, and is therefore more time-consuming than the computation of the spherical light wave scattered by a single point. However, complex waves scattered by scene points located within a single layer are calculated all at once using the Angular Spectrum. Therefore, this approach is more efficient than the point-source approach when the number of scene points within each layer is sufficiently important. However, when the scene geometry contains complex shapes, a large number of layers containing only one or a few points are needed to sample it, reducing the efficiency of the wave-field approach. In order to reduce the computational complexity of this approach, a technique that uses color-space conversion has been proposed [20].

These two approaches complement one another: while the point-source approach is very efficient when the scene contains complex shapes, such as trees or human bodies, the wave-field approach is more efficient when objects in the scene consist of large planar surfaces, such as roads or buildings. However, most real 3D scenes contain both complex shapes and planar surfaces. As a consequence, these two approaches are rarely fully efficient for computing CGH of real scenes in their entirety. In order to overcome these limitations, we propose a new CGH computation method with occlusion effect based on a fast hybrid point-source/wave-field approach. Whereas previously proposed methods tried to reduce the computational complexity of the point-source or the wave-field approaches independently, the proposed method uses the two approaches together and therefore takes advantages from both of them. Section 2 gives an overview of the method, Section 3 and Section 4 present respectively light propagation and light shielding techniques used by the method, and Section 5 gives the detailed algorithm of the method. Finally, Section 6 gives experimental results.

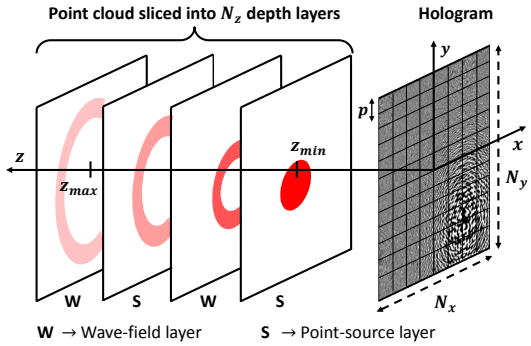

Figure 1: Scene geometry and coordinate system used by the proposed method.

2 Overview of the method

Figure 1 shows the scene geometry and coordinate system used by the proposed method. The coordinate system is defined by (x, y, z) so that the hologram lies on the $(x, y, 0)$ plane. The 3D scene is sliced into a set of N_z depth layers parallel to the hologram plane and located between $z_{\min}$ and $z_{\max}$. The distance between each layer is given by d_z. N_z is set such that the separation between two consecutive layers remains invisible [17]. The layers are numbered from 0 to $N_z - 1$, from the farthest to the nearest to the hologram plane. The layers and the hologram are sampled on a regular 2D grid of resolution $N \times N$ with sampling pitch p.

The layers are classified into two categories: the wave-field layers and the point-source layers. This classification is performed depending on the number of scene points within each layer: if the number of scene points M_d within layer d exceeds a threshold value $M_{d,\max}$, this layer is considered as a wave-field layer, otherwise it is considered as a point-source layer. The farthest and nearest layers are always considered as wave-field layers.

Each layer d operates as a surface source of light which emits a complex wave o_d given by

$$o_d(x,y) = A_d(x,y)\exp\left[j\phi_d(x,y)\right], \tag{1}$$

where $A_d(x,y)$ is the amplitude of the (x,y) point within layer d, calculated using illumination formulas, and $\phi_d(x,y)$ is its phase, set to a random value in order to render a diffusive scene. Light scattered by the scene is numerically propagated and shielded from one layer to another using the recurrence formula

$$\begin{aligned} &u_0(x,y) = o_0(x,y), \\ &u_d(x,y) = o_d(x,y) + \mathcal{O}_d\left\{\mathcal{P}_{d_z}\left\{u_{d-1}(x,y)\right\}\right\}, \end{aligned} \tag{2}$$

where u_d is the total complex wave scattered by layer d, operator $\mathcal{O}_d$ stands for light shielding by layer d, and operator $\mathcal{P}_z$ stands for the numerical propagation of light between two parallel planes separated by a distance z. These numerical operations are performed using either a point-source or a wave-field approach depending on which category layer d belongs to. Finally, the complex wave scattered by layer $N_z - 1$ is numerically propagated to the hologram plane in order to obtain the final CGH, according to

$$H(x,y) = \mathcal{P}_{z_{\min}}\left\{u_{N_z-1}(x,y)\right\}. \tag{3}$$

3 Light propagation

3.1 Wave-field layers

The propagation of complex waves scattered by wave-field layers is numerically computed using the Angular Spectrum [19], which expresses light propagation between two parallel planes separated by a distance z as

$$\mathcal{P}_z^w\left\{u_d(x,y)\right\} = \mathcal{F}^{-1}\Big\{\mathcal{F}\left\{u_d(x,y)\right\} \times \mathrm{e}^{-j2\pi\sqrt{\lambda^{-2}-f_x^2-f_y^2}z}\Big\}, \tag{4}$$

where λ is the wavelength of light, f_x and f_y are the spatial frequencies, and $\mathcal{F}$ is the forward Fourier Transform, which can be computed using the Fast Fourier Transform algorithm (FFT). The calculation of the right-hand side member of Eq. (4) has for computational complexity $O(N^2 log(N))$.

3.2 Point-source layers

Scene points located within point-source layers are considered as self-luminous points. The complex wave scattered by a point source k at coordinates (x_k, y_k, z) is given by the Angular Spectrum [19] as

$$w_k(x,y) = u_d(x_k,y_k)\mathcal{F}^{-1}\left\{\mathrm{e}^{-j2\pi\sqrt{\lambda^{-2}-f_x^2-f_y^2}z}\right\} \otimes \delta(x-x_k, y-y_k), \tag{5}$$

where $u_d(x_k,y_k)$ is the complex amplitude of the point and $\otimes$ is the convolution operator. In order to speed up the computation, we use a pre-calculated LUT, as proposed in [5]. The LUT is pre-computed as

$$T(x,y,z) = \mathcal{F}^{-1}\left\{\mathrm{e}^{-j2\pi\sqrt{\lambda^{-2}-f_x^2-f_y^2}z}\right\} h(x,y,z), \tag{6}$$

h being a window function used to restrict the region of contribution of a given point source, equal to one within the region of contribution of the point and zero elsewhere [4].

Then, light propagation between two parallel planes separated by a distance z is computed simply by addressing this pre-calculated LUT, as

$$\mathcal{P}_z^s\left\{u_d(x,y)\right\} = \sum_{k=0}^{M-1} u_d(x_k,y_k)T(x-x_k, y-y_k, z), \tag{7}$$

where M is the number of self-luminous points within the source plane. The calculation of the right-hand side member of Eq. (7) has for computational complexity $O(N^2M)$.

4 Light shielding

4.1 Wave-field layers

Light shielding by wave-field layers is performed using a binary silhouette mask function, as proposed in [16]. An occluding obstacle is located within layer d. The complex wave incident on layer d is given by

$$u'_d(x,y) = \mathcal{P}_{d_z}\left\{u_{d-1}(x,y)\right\}, \tag{8}$$

where u_{d-1} is the complex wave scattered by layer $d-1$. Part of u'_d is shielded by the occluding scene points, and its amplitude vanishes in the area of the obstacle. This is expressed by multiplying u'_d by a binary mask function m_d that has value zero inside the obstacle and one outside. Operator $\mathcal{O}^w_d$ is thus given by

$$\mathcal{O}^w_d\left\{u'_d(x,y)\right\} = m_d(x,y)u'_d(x,y). \tag{9}$$

If the scene points within layer d also emit a wave o_d, The complex wave incident on layer $d+1$ is given by

$$\begin{aligned} u'_{d+1}(x,y) &= \mathcal{P}_{d_z}\left\{o_d(x,y) + m_d(x,y)u'_d(x,y)\right\} \\ &= \mathcal{P}_{d_z}\left\{u_d(x,y)\right\}. \end{aligned} \tag{10}$$

While perfectly adapted to wave-field layers, this light shielding technique is not suited to point-source layers. Indeed, at each diffraction step, the complex wave scattered by the scene spreads gradually from one layer to another, and u'_d may spread out on a large number of samples. Thus, even if layer d contains only a few scene points, u_d may have a large number of non-zero values, which are therefore considered as a point sources by $\mathcal{P}^s_z$. Since the computational complexity of $\mathcal{P}^s_z$ is dependent on the number of point sources within the source layer, this light shielding technique is highly inefficient for point-source layers. In order to compute light shielding by point-source layers efficiently, we use a binary aperture function.

4.2 Point-source layers

Light shielding using a binary aperture function has been proposed by Matsushima *et al.* [21]. However, to the best of our knowledge, this technique has never been applied to point-source approach neither to CGH computation based on a layered model of the scene.

This time, we use an aperture function defined as

$$a_d(x,y) = 1 - m_d(x,y). \tag{11}$$

By substituting Eq. (11) in Eq. (10), u'_{d+1} becomes

$$\begin{aligned} u'_{d+1}(x,y) =& \mathcal{P}_{d_z}\big\{o_d(x,y) \\ &+ (1-a_d(x,y))\mathcal{P}_{d_z}\left\{u_{d-1}(x,y)\right\}\big\} \\ =& \mathcal{P}_{2d_z}\big\{u_{d-1}(x,y)\big\} + \mathcal{P}_{d_z}\big\{o_d(x,y) \\ &- a_d(x,y)\mathcal{P}_{d_z}\left\{u_{d-1}(x,y)\right\}\big\} \\ =& \mathcal{P}_{2d_z}\left\{u_{d-1}(x,y)\right\} + \mathcal{P}_{d_z}\left\{\hat{u}_d(x,y)\right\}, \end{aligned} \tag{12}$$

where the propagation $\mathcal{P}_{d_z}\left\{\mathcal{P}_{d_z}\left\{u_{d-1}(x,y)\right\}\right\}$ is equivalent to $\mathcal{P}_{2d_z}\left\{u_{d-1}(x,y)\right\}$, according to the definition of $\mathcal{P}_z$.

The computation of u'_{d+1} using the binary aperture involves using $\mathcal{P}_z$ three times, whereas we only need to use it twice using the binary mask. However, $\mathcal{P}_{d_z}\left\{u_{d-1}(x,y)\right\}$ needs to be calculated only within the region defined by a_d, which corresponds to the coordinates of the scene points located within layer d. Therefore, the number of non-zero values of $\hat{u}_d$ remains the same as the number of scene points. Using the binary aperture function has the great advantage of not increasing the number of point sources for which the complex wave has to be computed at each diffraction step. Operator $\mathcal{O}^s_d$ is thus given by

$$\mathcal{O}^s_d\left\{u'_d(x,y)\right\} = -a_d(x,y)u'_d(x,y). \tag{13}$$

5 Detailed algorithm

5.1 Point-source layers

Figure 2 shows the overall block-diagram of the proposed method. The complex wave incident on each point-source layer d is given by the sum of complex

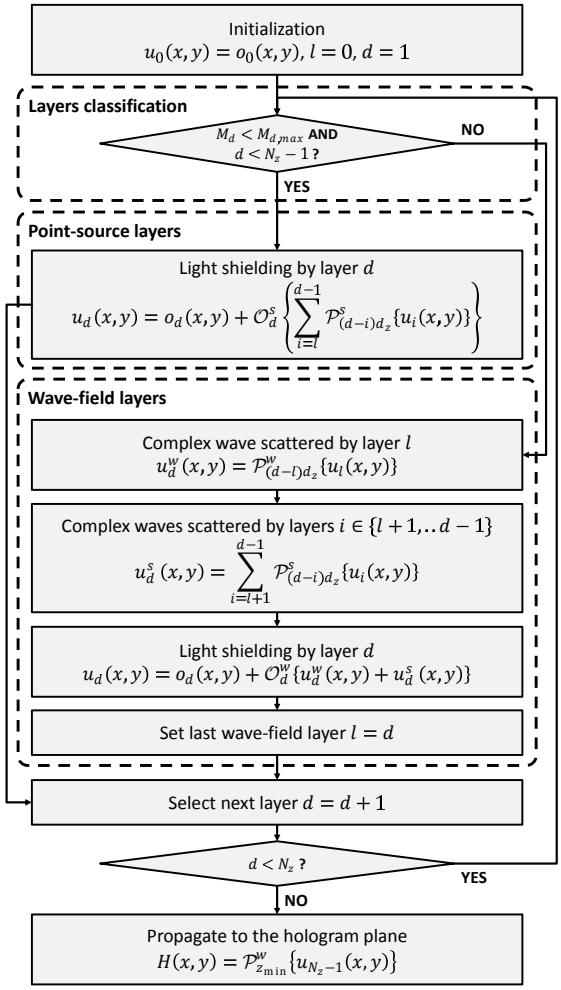

Figure 2: Block-diagram of the proposed method.

waves scattered by layers located between d and the last wave-field layer l. The propagation of these complex waves towards layer d is computed using $\mathcal{P}_z^s$ and shielded using $\mathcal{O}_d^s$. If the scene points within layer d also emit a wave o_d, the total complex wave scattered by this layer is given by

$$u_d(x,y) = o_d(x,y) - a_d(x,y) \sum_{i=l}^{d-1} \mathcal{P}_{(d-i)d_z}^s \left\{u_i(x,y)\right\}. \tag{14}$$

Since a_d has value one only at the coordinates of the scene points located within layer d, Eq. (14) needs to be calculated within this region only. The computational complexity of this step is thus $O(M_d M)$, where M is the total number of scene points within all the layers located between d and l: $M = \sum_{i=l}^{d-1} M_i$.

5.2 Wave-field layers

The complex wave incident on each wave-field layer d is given by the sum of the complex wave scattered by the last wave-field layer l and complex waves scattered by point-source layers located between d and l. The propagation of the complex wave scattered by the last wave-field layer l is computed using $\mathcal{P}_z^w$, according to

$$u_d^w(x,y) = \mathcal{P}_{(d-l)d_z}^w \left\{u_l(x,y)\right\}. \tag{15}$$

The computational complexity of this step is $O(N^2 log(N))$. By contrast, the propagation of the complex waves scattered by point-source layers is computed using $\mathcal{P}_z^s$, according to

$$u_d^s(x,y) = \sum_{i=l+1}^{d-1} \mathcal{P}_{(d-i)d_z}^s \left\{u_i(x,y)\right\}. \tag{16}$$

The computational complexity of this step is $O(N^2 M')$, where M' is the total number of scene points within the point-source layers located between d and l: $M' = \sum_{i=l+1}^{d-1} M_i$. If the scene points within layer d also emit a wave o_d, the total complex wave u_d scattered by this layer is given by

$$u_d(x,y) = o_d(x,y) + m_d(x,y)\left(u_d^w(x,y) + u_d^s(x,y)\right). \tag{17}$$

5.3 Layers classification

The first step to implement the proposed method is to determine the value of $M_{d,\max}$. We call t_d^s the time needed to compute light propagation and light shielding for layer d using the point-source approach,

(a) *Dice1* (b) *Dice2* (c) *City*

Figure 3: Computer graphics images of the three test scenes used for the experiments.

and t_d^w the time needed to compute it using the wave-field approach. t_d^s is given as the sum of the time needed to compute light shielding by layer d using $\mathcal{O}_d^s$ and the time needed to compute the propagation of complex wave scattered by layer d using $\mathcal{P}_z^s$:

$$t_d^s = \alpha M_d M + \beta M_d N^2, \tag{18}$$

where α and β are constant coefficients. t_d^w is given by

$$t_d^w = \gamma N^2 log(N), \tag{19}$$

where γ is a constant. We found the numerical values for the coefficients α, β and γ using the Gnuplot implementation of the nonlinear least-squares Levenberg-Marquardt algorithm [22]:

$$\alpha = 6,37.10^{-8}, \quad \beta = 1,07.10^{-7}, \quad \gamma = 5,35.10^{-8}. \tag{20}$$

In order to maximize the efficiency of the proposed method, $M_{d,\max}$ must be set such that

$$t_d^s = t_d^w \Leftrightarrow M_{d,\max} = \frac{\gamma N^2 log(N)}{\alpha M + \beta N^2}. \tag{21}$$

6 Experimental results

The proposed method was implemented in C++/CUDA on a PC system employing an Intel Core i7-4930K CPU operating at 3.40 GHz, a main memory of 16 GB as well as three GPUs NVIDIA GeForce GTX 780Ti. For the experiments, we used three different test scenes (cf. Figure 3).

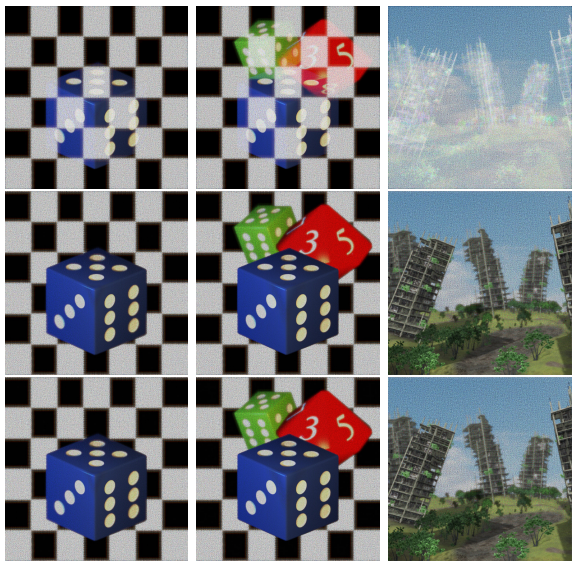

Figure 4: Numerical reconstructions of the CGHs generated by the point-source method (first row), the wave-field method (second row), and the proposed method (third row).

The scenes are sliced into a set of N_z depth layers parallel to the hologram plane and located between $z_{\min} = 0$ and $z_{\max} = 2$cm. N_z is set to 512, 1024 and 2048 for *Dice1*, *Dice2* and *City*, respectively. The layers and the hologram are sampled on a regular 2D grid of resolution 4096×4096 with a sampling pitch $p = 8.1\mu$m. We compared the proposed method with GPU implementations of two other methods: (1) the point-source method without occlusion effect proposed in [5], and (2) the wave-field method with occlusion effect proposed in [16]. We adapted both methods to produce colorful complex modulation CGH.

Figure 4 shows the scene images numerically reconstructed from the CGH patterns of the three test scenes generated by the point-source method (first row), the wave-field method (second row), and the proposed method (third row). As seen in first row, the point-source method does not take into account occlusions in the scene, and the objects appear therefore semi-transparent, strongly limiting the realism of the displayed image. On the other hand, occlu-

Scene	Number of layers	Number of scene points	Method	Calculation time		
				Total	Per layer	Per point
Dice1	512	12,820,048	Wave-field method	6.77s (100%)	13.22ms	0.53μs
			Point-source method	7.28s (108%)	14.22ms	0.57μs
			Proposed method	5.01s (74.0%)	9.79ms	0.39μs
Dice2	1024	16,736,640	Wave-field method	13.97s (100%)	13.64ms	0.83μs
			Point-source method	9.71s (69.5%)	9.48ms	0.58μs
			Proposed method	8.75s (62.6%)	8.54ms	0.52μs
City	2048	47,648,608	Wave-field method	26.88s (100%)	13.12ms	0.56μs
			Point-source method	25.74s (95.8%)	12.57ms	0.54μs
			Proposed method	19.17s (71.3%)	9.36ms	0.40μs

Table 1: CGH computation times for the three test scenes using the wave-field, point-source and proposed methods.

sions between objects in the scene are accurately reproduced using the wave-field and proposed methods without producing any visible artifact (rows 2 and 3).

As shown in Table 1, using the wave-field method, the average CGH calculation time per layer remains constant for the three test scenes. This means that the total calculation time increases linearly with the number of layers and is not dependent on the total number of scene points. By contrast, using the point-source method, the average CGH calculation time per scene point remains constant for the three test scenes. This means that the total calculation time increases linearly with the number of scene points and is not dependent on the number of layers. As a result, the point-source method is less efficient than the wave-field method when the number of points per layer is high, as in scene *Dice1*, and is more efficient when the average number of points within each layer is low, as in scenes *Dice2* and *City*.

As shown in Table 1, the total calculation time using the proposed method is dependent both on the number of layers and on the number of scene points. By combining these two approaches, the proposed method takes advantages from both of them and is therefore always more efficient. Using the proposed method, the total CGH calculation time has been decreased by 26.0%, 37.4% and 28.7% for *Dice1*, *Dice2* and *City*, respectively, compared to the wave-field method, and by 31.2%, 9.89% and 25.5% for *Dice1*, *Dice2* and *City*, respectively, compared to the point-source method. These experimental results confirm the performance superiority of the proposed method over the point-source and wave-field methods in terms of computation time.

7 Conclusion

In this paper, we proposed a fast CGH computation method with occlusion effect based on a hybrid point-source/wave-field approach. The 3D scene is sliced into several depth layers parallel to the hologram plane. Then, light scattered by the scene is propagated and shielded from one layer to another using either a point-source or a wave-field approach according to a threshold criterion on the number of points within the layer. Finally, we compute light propagation from the nearest layer to the hologram plane in order to obtain the final CGH.

Experimental results reveal that our method accurately takes into account occlusions between objects in the scene without producing any visible artifact. Furthermore, the CGH calculation time has been reduced up to 31,2% and 37.4% compared to the point-source and wave-field methods, respectively. This confirms the performance superiority of our method over the point-source and wave-field approaches in terms of computation time.

References

[1] Ulf Schnars and Werner Jüptner. *Digital Holography: Digital Hologram Recording, Numerical Reconstruction, and Related Techniques*. Springer Science & Business Media, December 2005.

[2] John S. Underkoffler. Occlusion processing and smooth surface shading for fully computed synthetic holography. In *Practical Holography XI and Holographic Materials III*, volume Proc. SPIE 3011, pages 19–30, April 1997.

[3] Hao Zhang, Qiaofeng Tan, and Guofan Jin. Full parallax three-dimensional computer generated hologram with occlusion effect using ray casting technique. *Journal of Physics: Conference Series*, 415(1):012048, February 2013.

[4] Mark E. Lucente. Interactive computation of holograms using a look-up table. *Journal of Electronic Imaging*, 2(1):28–34, January 1993.

[5] Seung-Cheol Kim and Eun-Soo Kim. Effective generation of digital holograms of three-dimensional objects using a novel look-up table method. *Applied Optics*, 47(19):D55–D62, July 2008.

[6] Hiroshi Yoshikawa, Susumu Iwase, and Tadashi Oneda. Fast computation of Fresnel holograms employing difference. In *Practical Holography XIV and Holographic Materials VI*, volume Proc. SPIE 3956, pages 48–55, May 2000.

[7] Kyoji Matsushima and Masahiro Takai. Recurrence Formulas for Fast Creation of Synthetic Three-Dimensional Holograms. *Applied Optics*, 39(35):6587–6594, December 2000.

[8] Takeshi Yamaguchi, Gen Okabe, and Hiroshi Yoshikawa. Real-time image plane full-color and full-parallax holographic video display system. *Optical Engineering*, 46(12):125801–125801–8, 2007.

[9] Tomoyoshi Shimobaba, Hirotaka Nakayama, Nobuyuki Masuda, and Tomoyoshi Ito. Rapid calculation algorithm of Fresnel computer-generated-hologram using look-up table and wavefront-recording plane methods for three-dimensional display. *Optics Express*, 18(19):19504–19509, September 2010.

[10] Jiantong Weng, Tomoyoshi Shimobaba, Naohisa Okada, Hirotaka Nakayama, Minoru Oikawa, Nobuyuki Masuda, and Tomoyoshi Ito. Generation of real-time large computer generated hologram using wavefront recording method. *Optics Express*, 20(4):4018–4023, February 2012.

[11] Lukas Ahrenberg, Philip Benzie, Marcus Magnor, and John Watson. Computer generated holography using parallel commodity graphics hardware. *Optics express*, 14(17):7636–7641, August 2006.

[12] Nobuyuki Masuda, Tomoyoshi Ito, Takashi Tanaka, Atsushi Shiraki, and Takashige Sugie. Computer generated holography using a graphics processing unit. *Optics Express*, 14(2):603–608, January 2006.

[13] Yasuyuki Ichihashi, Hirotaka Nakayama, Tomoyoshi Ito, Nobuyuki Masuda, Tomoyoshi Shimobaba, Atsushi Shiraki, and Takashige Sugie. HORN-6 special-purpose clustered computing system for electroholography. *Optics Express*, 17(16):13895–13903, August 2009.

[14] Rick H.-Y. Chen and Timothy D. Wilkinson. Computer generated hologram from point cloud using graphics processor. *Applied Optics*, 48(36):6841–6850, December 2009.

[15] Ivo Hanák, Adam Herout, and Pavel Zemčík. Acceleration of detail driven method for hologram generation. *Optical Engineering*, 49(8):085802–085802–9, 2010.

[16] A. W. Lohmann. Three-dimensional properties of wave-fields. *Optik*, 51:105–107, 1978. 00065.

[17] Muharrem Bayraktar and Meriç Özcan. Method to calculate the far field of three-dimensional objects for computer-generated holography. *Applied Optics*, 49(24):4647–4654, August 2010.

[18] Ryutaro Oi, Kenji Yamamoto, and Makoto Okui. Electronic generation of holograms by using depth maps of real scenes. volume 6912, pages 69120M–69120M–11, 2008.

[19] Joseph W. Goodman. *Introduction to Fourier Optics*. Roberts and Company Publishers, Englewood, Colo, 3rd edition, 2005.

[20] Tomoyoshi Shimobaba, Takashi Kakue, and Tomoyoshi Ito. Acceleration of color computer-generated hologram from three-dimensional scenes with texture and depth information. volume 9117, pages 91170B–91170B–8, 2014.

[21] Kyoji Matsushima, Masaki Nakamura, and Sumio Nakahara. Silhouette method for hidden surface removal in computer holography and its acceleration using the switch-back technique. *Optics Express*, 22(20):24450–24465, October 2014.

[22] D. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963. 22862.